



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/757,517	01/10/2001	Stephen E. Fischer	FIS920000280US1	8455

29505 7590 12/23/2003

DELIO & PETERSON, LLC
121 WHITNEY AVENUE
NEW HAVEN, CT 06510

EXAMINER

FOWLKES, ANDRE R

ART UNIT	PAPER NUMBER
----------	--------------

2122

DATE MAILED: 12/23/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/757,517

Applicant(s)

FISCHER, STEPHEN E.

Examiner

Andre R. Fowlkes

Art Unit

2122

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 03 May 2001.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-15 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-15 is/are rejected.
- 7) ☒ Claim(s) 5 is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☒ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 10 January 2001 is/are: a) ☐ accepted or b) ☒ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. §§ 119 and 120

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
* See the attached detailed Office action for a list of the certified copies not received.
- 13) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application) since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.
a) ☐ The translation of the foreign language provisional application has been received.
- 14) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121 since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892) 4) ☐ Interview Summary (PTO-413) Paper No(s). _____
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948) 5) ☐ Notice of Informal Patent Application (PTO-152)
- 3) ☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s) 2. 6) ☐ Other: _____

DETAILED ACTION

1. Claims 1-15 are pending.

Drawings

2. New corrected drawings are required in this application because it is noted that the label numbering is informal. Applicant is advised to employ the services of a competent patent draftsman outside the Office, as the U.S. Patent and Trademark Office no longer prepares new drawings. The corrected drawings are required in reply to the Office action to avoid abandonment of the application. The requirement for corrected drawings will not be held in abeyance.

Specification

3. The disclosure is objected to because of the following informalities:
 - "provides a utility program" should be "provides a utility program that" on p. 6, lines 24, 26 and p. 7, lines 1, 6;
 - "associated with given" should be "associated with a given" on p. 16, line 18;
 - "can be create via" should be "can be created via" on p. 17, line 3;Appropriate correction is required.

Claim Objections

4. Claim 5 is objected to because of the following informalities:

- "program including scripting language specifying" should be "program including a scripting language specifying" in claim 5, lines 7-8;

Appropriate correction is required.

Claim Rejections - 35 USC § 112

5. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claim 12 contains the trademark/trade name UNIX®. Where a trademark or trade name is used in a claim as a limitation to identify or describe a particular material or product, the claim does not comply with the requirements of 35 U.S.C. 112, second paragraph. See *Ex parte Simpson*, 218 USPQ 1020 (Bd. App. 1982). The claim scope is uncertain since the trademark or trade name cannot be used properly to identify any particular material or product. A trademark or trade name is used to identify a source of goods, and not the goods themselves. Thus, a trademark or trade name does not identify or describe the goods associated with the trademark or trade name. In the present case, the trademark/trade name is used to identify/describe an operating system environment and, accordingly, the identification/description is indefinite.

Claim Rejections - 35 USC § 102

6. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1 - 3, 14, and 15 are rejected under 35 U.S.C. 102(b) as being anticipated by the IBM Technical Disclosure Bulletin (IBM TDB), "Visual Make Utility", Volume 39, No. 02, February 1996 (Art of Record).

As per claim 1, the IBM TDB discloses:

- A method for updating existing code in a computer program after inputting new code which defines changes to said existing code (p. 363, ¶ 1, lines 3 – 5, "The makefile can be used in conjunction with a make utility to automatically determine which target files are out of date with respect to their dependant files, and therefore rebuild the targets which are out of date").

- generating a target file list which includes target files (p. 363, ¶ 1, lines 1 – 2, "Make files are used by programmers to determine a set (list) of target and dependant files").

- generating an associated file list including associated files which correspond with said target files (p. 363, ¶ 1, lines 1 – 2, "Make files are used by programmers to determine a set (list) of target and dependant (associated) files").

- executing an algorithm where said algorithm locates said target files by employing a search process, said algorithm updates said target files and updates said associated files by selectively compiling said target files (p. 363, ¶ 1, lines 1 – 5, "Make files are used by programmers to determine (search and locate) a set of target and dependant files. The makefile can be used in conjunction with a make

utility to automatically determine (using an algorithm) which target files are out of date with respect to their dependant files, and therefore rebuild (update) the targets which are out of date”).

As per claim 2, the rejection of claim 1 is incorporated and further the IBM TDB discloses **source code and object code, said target files being source code and said associated files being object code, said source code being selectively compiled to update and provide said associated object code** (p. 365, figure 3 shows a collection of source code and object code. Additionally, p. 363, ¶ 1, lines 3-5, “The makefile can be used in conjunction with a make utility to automatically determine which target files are out of date with respect to their dependant files, and therefore rebuild the targets which are out of date”).

As per claim 3, the rejection of claim 1 is incorporated and further the IBM TDB discloses **updating said associated file list with new said associated files, said new associated files being defined by said new code** (p. 363, ¶ 1, lines 3 – 5, “The makefile can be used in conjunction with a make utility to automatically determine which target files are out of date with respect to their dependant files, and therefore rebuild the targets which are out of date”).

As per claim 14, the IBM TDM discloses:

- A computer program product for updating existing code in a computer program after inputting new code which defines changes to said existing code, said computer program product (p. 363, ¶ 1, lines 3 – 5, “The makefile can be used in conjunction with a make utility to automatically determine which target files are out of date with respect to their dependant files, and therefore rebuild the targets which are out of date”).

- computer readable program code means for generating a target file list which includes target files (p. 363, ¶ 1, lines 1 – 2, “Make files are used by programmers to determine a set (list) of target and dependant files”, and p. 364, ¶ 1, line 3, discusses a makefile used in an “OS/2” implementation, therefore, the makefile discussed is inherently computer readable program code).

- computer readable program code means for generating an associated file list including associated files which correspond with said target files (p. 363, ¶ 1, lines 1 – 2, “Make files are used by programmers to determine a set (list) of target and dependant files”, and p. 364, ¶ 1, line 3, discusses a makefile used in an “OS/2” implementation, therefore, the makefile discussed is inherently computer readable program code).

- computer readable program code means for executing an algorithm where said algorithm locates said target files by employing a search process, said algorithm updates said target files and updates said associated files by selectively compiling said target files (p. 363, ¶ 1, lines 3 – 5, “The makefile can be used in conjunction with a make utility to automatically determine (search and locate)

which target files are out of date with respect to their dependant files, and therefore rebuild the targets which are out of date”).

As per claim 15, the IBM TDB discloses:

- a program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps for updating existing code in a computer program after inputting new code which defines changes to said existing code (p. 364, ¶ 1, line 3, discusses a makefile used in an “OS/2” implementation, and OS/2 is a computer operating system (program of instructions) that is stored and used on computer systems (that contain program storage devices). Additionally, p. 363, ¶ 1, lines 3 – 5, “The makefile can be used ... (to) rebuild the targets (code) which are out of date”).

- generating a target file list which includes target files (p. 363, ¶ 1, lines 1 – 2“, Make files are used by programmers to determine a set (list) of target and dependant files”).

- generating an associated file list including associated files which correspond with said target files (p. 363, ¶ 1, lines 1 – 2“, Make files are used by programmers to determine a set (list) of target and dependant (associated) files”).

- executing an algorithm where said algorithm locates said target files by employing a search process, said algorithm updates said target files and updates said associated files by selectively compiling said target files (p. 363, ¶ 1, lines 3 – 5, “The makefile can be used in conjunction with a make utility to automatically

determine (search and locate) which target files are out of date with respect to their dependant files, and therefore rebuild the targets which are out of date", and since the makefile selectively rebuilds the target files, it inherently uses an algorithm for the search and selection process).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

7. Claims 4 – 10 and 12 are rejected under 35 U.S.C. 103(a) as being unpatentable over the IBM Technical Disclosure Bulletin (IBM TDB), "Visual Make Utility", Volume 39, No. 02, February 1996 (Art of Record), in view of Auer et al. (Auer), U.S. Patent No. 6,067,637.

As per claim 4, the rejection of claim 1 is incorporated and further, the IBM TDB doesn't explicitly disclose **including pattern type variables which use generic rules to specify said associated object code for updating.**

However, Auer, in an analogous environment, discloses a system for **including pattern type variables which use generic rules to specify data for updating** (col. 12 lines 31-35, "If the regular expression pattern and the variables in the 'pattern' match the current line of the current screen ... evaluate the conditional expression ...

substitute a matched value from the working memory match for each ... variable in the 'condition' expression. Substitute a matched value from the screen match for each \$I variable in the 'condition' expression").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Auer into the system of the IBM TDB to have **pattern type variables which use generic rules to specify associated data for updating**. The modification would have been obvious because one of ordinary skill in the art would want use to pattern type variables to efficiently locate the data for updating while minimizing the amount of time and operating system resources used.

As per claim 5, the IBM TDB discloses:

- **A method for generating changes and updating existing files and code in a computer program** (p. 363, ¶ 1, lines 3 – 5, "The makefile can be used in conjunction with a make utility to automatically determine which target files are out of date with respect to their dependant files, and therefore rebuild the targets which are out of date").

- **reading existing source code and existing object code in said computer program** (p. 363, ¶ 3, lines 1 - 2, "Another component of the typical make utility is a tool that can parse source files to determine their dependent files").

- **reading a plurality of associated files where said associated files are associated with said source code** (p. 363, ¶ 3, line 1, "Another component of the typical make utility is a tool that can parse ... files").

-executing a utility program which updates target said source code and said object code associated with said target source code (p. 363, ¶ 1, lines 3 – 5, “The makefile can be used in conjunction with a make utility to automatically determine which target files are out of date with respect to their dependant files, and therefore rebuild the targets which are out of date”).

- generating a target code list for said source code and said associated object code by using said utility program (p. 363, ¶ 1, lines 3 – 4, “The makefile can be used in conjunction with a make utility to automatically determine which target files are out of date”).

The IBM TDB doesn't explicitly disclose the **utility program including (a) scripting language specifying particular characters to search for in said target code and said associated code and updating said target code and said associated code by employing a search technique defined in said utility program, said search technique includes specified patterns such that said specified patterns identify said existing associated code being updated.**

However, Auer, in an analogous environment, discloses the **utility program including scripting language specifying particular characters to search for in a file** (col. 2 lines 27 – 30, “The expert system (scripter/interpreter) preferably executes an action ... only if the comparator determines that the match portion of the one of the event triggers (particular characters to search for) matches information received”), and **updating said file by employing a search technique defined in said utility program, said search technique includes specified patterns such that said specified**

patterns identify said file being updated (col. 12 lines 31 – 35, “If the regular expression pattern and the variables in the ‘pattern’ match the current line of the current screen ... evaluate the conditional expression ... substitute (update) a matched value from the working memory match for each ... variable in the ‘condition’ expression. . . Substitute a matched value from the screen match for each \$I variable in the ‘condition’ expression”).

Therefore, it would have been obvious to a person of ordinary skill in the art, at the time the invention was made, to incorporate the teachings of Auer into the system of the IBM TDB to have the **utility program including scripting language specifying particular characters to search for in said target code and said associated code and updating said target code and said associated code by employing a search technique defined in said utility program, said search technique includes specified patterns such that said specified patterns identify said existing associated code being updated.** The modification would have been obvious because one of ordinary skill in the art would want to use to a scripting language and pattern type variables to efficiently locate the data for updating while minimizing the amount of time and operating system resources used.

As per claim 6, the rejection of claim 5 is incorporated and further the IBM TDB doesn't explicitly disclose that **specified patterns of said search technique includes pattern type variables which use generic rules to specify said associated object code for updating.**

However, Auer, in an analogous environment, discloses a system where **specified patterns of said search technique includes pattern type variables which use generic rules to specify data for updating** (col. 12 lines 31-35, "If the regular expression pattern and the variables in the 'pattern' match the current line of the current screen ... evaluate the conditional expression ... substitute a matched value from the working memory match for each ... variable in the 'condition' expression. Substitute a matched value from the screen match for each \$I variable in the 'condition' expression").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Auer into the system of the IBM TDB so that **specified patterns of said search technique includes pattern type variables which use generic rules to specify said associated object code for updating**. The modification would have been obvious because one of ordinary skill in the art would want use to pattern type variables to efficiently locate data for updating while minimizing the amount of time and operating system resources used.

As per claim 7, the rejection of claim 5 is incorporated, and further the IBM TDB doesn't explicitly disclose **matching specified characters in said target code and said associated code such that said specified characters identify said existing associated code being updated**.

However, Auer, in an analogous environment, discloses the **matching specified characters in a file such that said specified characters identify said existing file**

Art Unit: 2122

being updated (col. 2 lines 27 – 30, “The expert system preferably executes an action ... only if the comparator determines that the match portion of the one of the event triggers (particular characters to search for) matches information received”)

Therefore, it would have been obvious to a person of ordinary skill in the art, at the time the invention was made, to incorporate the teachings of Auer into the system of the IBM TDB, to provide **matching specified characters in said target code and said associated code such that said specified characters identify said existing associated code being updated**. The modification would have been obvious because one of ordinary skill in the art would want to match specified characters in order to efficiently locate the file for updating while minimizing the amount of time and operating system resources used.

As per claim 8, the IBM TDB discloses:

- **A method for generating changes and updating existing source code and existing object code in a computer program which includes a plurality of dependent files which are prerequisites of said source files** (p. 363, ¶ 1 lines 3 - 5, “The makefile can be used in conjunction with a make utility to automatically determine which target files are out of date with respect to their dependant files, and therefore rebuild the targets which are out of date”).

- **inputting a utility program which updates target said source code, said object code being associated with said target source code, and said utility program** (p. 363, ¶ 1 lines 3 - 5, “The makefile can be used in conjunction with a make

Art Unit: 2122

utility to automatically determine which target files are out of date with respect to their dependant files, and therefore rebuild the targets which are out of date”).

- **generating a target code list for changing said associated object code using said utility program** (p. 363, ¶ 2 lines 4 – 6, “It specifies (lists) ... target files ... (and) dependant files are listed for each target).

- **updating said associated object code** (p. 363, ¶ 1 lines 3 - 5, “The makefile can be used in conjunction with a make utility to automatically determine which target files are out of date with respect to their dependant files, and therefore rebuild the targets which are out of date”).

The IBM TDB doesn't explicitly disclose **including a scripting language specifying particular characters to search for in said target code and said associated object code or executing a search technique including pattern type variables which use generic rules to specify said associated object code for changing.**

However, Auer, in an analogous environment, discloses **including a scripting language specifying particular characters to search for in a file** (col. 2 lines 27 – 30, “The expert system (scripter/interpreter) preferably executes an action ... only if the comparator determines that the match portion of the one of the event triggers (particular characters to search for) matches information received”) and **executing a search technique including pattern type variables which use generic rules to specify said data for changing** (col. 12 lines 31-35, “If the regular expression pattern and the variables in the 'pattern' match the current line of the current screen ... evaluate the

conditional expression ... substitute a matched value from the working memory match for each ... variable in the 'condition' expression. Substitute a matched value from the screen match for each \$I variable in the 'condition' expression").

Therefore, it would have been obvious to a person of ordinary skill in the art, at the time the invention was made, to incorporate the teachings of Auer into the system of the IBM TDB, **including a scripting language specifying particular characters to search for in said target code and said associated code and executing a search technique including pattern type variables which use generic rules to specify said associated code for changing.** The modification would have been obvious because one of ordinary skill in the art would want to use to a scripting language and pattern type variables to efficiently locate the data for updating while minimizing the amount of time and operating system resources used.

As per claim 9, the rejection of claim 8 is incorporated and further, the IBM TDB discloses that the **utility program defines new source code to be added to said existing source code** (p. 363, ¶ 1 lines 3 - 5, "The makefile can be used in conjunction with a make utility to automatically determine which target files are out of date with respect to their dependant files, and therefore rebuild (define new code to be added to) the targets which are out of date").

As per claim 10, the rejection of claim 8 is incorporated and further, the IBM TDB discloses that **the utility program prioritizes said target code to update while**

employing said search technique (p. 366, ¶ 2 lines 1 – 2, “The cpp and hpp files could be added to the hierarchy (a technique used to prioritize code to update) automatically by executing the make dependency command on higher nodes in the tree”).

As per claim 12, the rejection of claim 8 is incorporated and further, the IBM TDM doesn't explicitly disclose that the **utility program is in a UNIX® environment**.

However, Auer, in an analogous environment, discloses a **program that is in a UNIX® environment** (col. 3 lines 16 - 17, “executes an expert system (program) under the UNIX® operating system”).

Therefore, it would have been obvious to a person of ordinary skill in the art, at the time the invention was made, to incorporate the teachings of Auer into the system of the IBM TDB to have the utility program in a UNIX® environment. The modification would have been obvious because one of ordinary skill in the art would be motivated to have the program in a UNIX® environment so that the program is highly portable and available to use on many machines.

8. Claims 11 and 13 are rejected under 35 U.S.C. 103(a) as being unpatentable over the IBM Technical Disclosure Bulletin (IBM TDB), “Visual Make Utility”, Volume 39, No. 02, February 1996 (Art of Record), in view of Auer et al. (Auer), U.S. Patent No. 6,067,637, in further view of Safonov, U.S. Patent No. 5,892,951.

As per claim 11, the rejection of claim 8 is incorporated and further, the IBM TDB doesn't explicitly disclose that the **utility program includes a process procedure for an operator to call, said process procedure recursively invokes said utility.**

However, Safonov, in an analogous environment, discloses that the **utility program includes a process procedure for an operator to call, said process procedure recursively invokes said utility** (col. 2, lines 32-33, describes a situation that "requires the compiler to do a recursive descent through (a representation of the source code)").

Therefore, it would have been obvious to a person of ordinary skill in the art, at the time the invention was made, to incorporate the teachings of Safonov into the system of the IBM TDB so that the **utility program includes a process procedure for an operator to call, said process procedure recursively invokes said utility.** The modification would have been obvious because one of ordinary skill in the art would want to create the procedure using recursive techniques in order to create a simple but powerful procedure with a minimal amount of coding.

As per claim 13, the rejection of claim 8 is incorporated and further the IBM TDB doesn't explicitly disclose that the **utility program provides that said existing code with a specific character are not considered to be a file, and thereby are bypassed for any changes by the utility program**

However, Safonov, in an analogous environment, discloses that the **utility program provides that said existing code with a specific character are not**

Art Unit: 2122

considered to be a file, and thereby are bypassed for any changes by the utility program (col. 1, lines 55 – 58, “a conventional compiler may include a lexical analyzer that looks at the source program and identifies successive ‘tokens’ (specific characters) in the source program” and these tokens are used to denote files that are to be bypassed for any changes).

Therefore, it would have been obvious to a person of ordinary skill in the art, at the time the invention was made, to incorporate the teachings of Safonov into the system of the IBM TDB so that the **utility program provides that said existing code with a specific character are not considered to be a file, and thereby are bypassed for any changes by the utility program** The modification would have been obvious because one of ordinary skill in the art would want to reduce processing time by ignoring files that don't need to be modified.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Andre R. Fowlkes whose telephone number is (703)305-8889. The examiner can normally be reached on Monday - Friday, 8:00am-4:30pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (703)305-4552. The fax phone number for the organization where this application or proceeding is assigned is (703) 872-9306.

Art Unit: 2122

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703)305-3900.

ARF

Wei Zhen
WEI ZHEN
Primary Examiner